
**NLSY97 Appendix 3: Family Background
and Formation Variable Creation**

HOUSEHOLD SIZE AS OF THE SURVEY DATE

Variables Created: CV_HH_SIZE
CV_HH_UNDER_6
CV_HH_UNDER_18

Variables Used

Name in Program	Question Name on CD
hage01-hage14	HHI_AGE.01-.14
huid01-huid14	HHI_UID.01-.14
id	PUBID
rsage	SYMBOL!KEY!AGE

This program creates several variables describing the composition of the respondent's household: the total number of residents, the number of residents under age 6, and the number of residents under age 18.

```

array hage hage01-hage14;
array huid huid01-huid14;
array hhdum hhdum01-hhdum14;
array dum6 dum601-dum614;
array dum18 dum1801-dum1814;

do i=1 to 14;
  hhdum[i]=0; dum6[i]=0; dum18[i]=0;
end;

do i=1 to 14;
  if huid[i]>-1 then hhdum[i]=1;
  if -1<hage[i]<6 then dum6[i]=1;
  if -1<hage[i]<18 then dum18[i]=1;
end;

/* create the dummy for the respondent's age rdum6
   and rdum18. */
rdum6=0;
rdum18=0;

if -1<rsage<6 then rdum6=1;
if -1<rsage<18 then rdum18=1;

/* Create household size hhsiz by adding up the
   dummies hhdum[i] and also add one due to the
   respondent himself. Similarly, create variables
   under6 and under18 by adding up the other two
   dummies.*/
hhsiz=1;
under6=rdum6;
under18=rdum18;

do i=1 to 14;
  hhsiz=hhsiz+hhdum[i];
  under6=under6+dum6[i];
  under18=under18+dum18[i];
end;

if huid01=-5 then hhsiz=-5;
if -4<huid01<0 or -4<huid02<0 or -4<huid03<0 or
   -4<huid04<0 or -4<huid05<0 or -4<huid06<0 or
   -4<huid07<0 or -4<huid08<0 or -4<huid09<0 or
   -4<huid10<0 or -4<huid11<0 or -4<huid12<0 or
   -4<huid13<0 or -4<huid14<0 then hhsiz=-3;

if huid01=-5 then under6=-5;
if huid01=-5 then under18=-5;
if -4<hage01<0 or -4<hage02<0 or -4<hage03<0 or
   -4<hage04<0 or -4<hage05<0 or -4<hage06<0 or
   -4<hage07<0 or -4<hage08<0 or -4<hage09<0 or
   -4<hage10<0 or -4<hage11<0 or -4<hage12<0 or
   -4<hage13<0 or -4<hage14<0 or -4<rsage<0 then
  do;
    under6=-3;
    under18=-3;
  end;

/* this respondent has household uid02 eq to -3 but has
   hhage02 eq to 40. created variable is correct to
   avoid -3 */
if id = 1820961 then hhsiz = 8;

endsas;

```

YOUTH'S RELATIONSHIP TO HOUSEHOLD PARENT FIGURE(S)

Variables Created: CV_YTH_REL_HH_CURRENT

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
Round 1:		Round 4:	
pubid	PUBID	hhage01-14	HHI_AGE.01-.14
yid	YOUTH_ID.01	hh2uid01-14	HHI_UID.01-.14
sh931-sh935	SH-93.01-.05	marry_1-14	HHI_MARSTAT.01-.14
hh1uid01-hh1uid16	HHI2_UID.01-.16	rel_1-14	HHI_REL.01-.14
		age	SYMBOL!KEY!AGE

Codes for Created Variable

- 5 = Non-Interview
- 3 = Invalid Skip
- 1 = Both biological parents
- 2 = Two parents, biological mother
- 3 = Two parents, biological father
- 4 = Biological mother only
- 5 = Biological father only
- 6 = Adoptive parent(s)
- 7 = Foster parent(s)
- 8 = No parents, grandparents
- 9 = No parents, other relatives
- 10 = Anything else

This program creates a variable identifying the youth's relationship to the primary adults in the household. In round 4 (as in rounds 2 and 3) there was no information collected on the legal guardian of the youth, so it is not possible to determine whether respondents are living with non-parent relatives because they are guardians or because the living situation is better (e.g., closer to school, no rent). For this reason, youths above the age 18 and above are considered independent and, if they are not living with an identified parent or parent-figure (legal guardian), are put into the anything else category. Youths below the age of 18 who are not living with an identified parent or parent-figure are put into the category that most closely matches their household situation.

```
array rel_a (i) rel_1-rel_14;
array gend_a (i) gend_1-gend_14;
array marry_a (i) marry_1-marry_14;
array age_a (i) hhage01-hhage14;
```

*this part determines legal guardians so that we can determine whether any guardians are present from the R1 interview use only unique IDs (cases with 6 digits are hh members after R1 only, so this program won't do anything with them, as we don't have info on legal guardian status for these HH members ;

```
array hh1uid (i) hh1uid01-hh1uid14; *round 1 variable;
array hh2uid (i) hh2uid01-hh2uid14; *round 4 variable;
```

```
do i=1 to 14;
  if yid=1 and sh931>0 and sh931+100=hh1uid then do; uid=hh1uid; end;
  if yid=2 and sh932>0 and sh932+100=hh1uid then do; uid=hh1uid; end;
  if yid=3 and sh933>0 and sh933+100=hh1uid then do; uid=hh1uid; end;
  if yid=4 and sh934>0 and sh934+100=hh1uid then do; uid=hh1uid; end;
  if yid=5 and sh935>0 and sh935+100=hh1uid then do; uid=hh1uid; end;
end;
do i=1 to 14; if (uid ne . and uid=hh2uid) then line=i; end;
```

```
legal=0;
do i=1 to 14; if line=i then legal=rel_a; end;
momid=0; domid=0; adopdad=0; admom=0; fostma=0; fostda=0;
stepma=0; stepda=0; husb=0; wife=0; grand=0; relat=0;
nonrel=0; indep=0; spouse=0;
```

```
**legal;
do i=1 to 14;
  if (legal>28 and legal<37) then do; grand=1; end;
```

Appendix 3: Family Background and Formation Variable Creation

```
*spouse;          if legal=1 or legal=2 then do; spouse=1; end;
*brother/sister;  if (legal>12 and legal<19) and age_a>20 then do; relat=1; end;
*aunt/uncle/other rel.; if (legal>69 and legal<85) and age_a>20 then do; relat=1; end;
*lover, roommate, other non-relative, mom's or dad's partner;
if legal=69 or legal=68 or legal=85 or legal=88 or legal=89 then do; nonrel=1; end;
if legal=-1 or legal=-2 or legal=-3 then do; invalid=1; end;
if rel_1=-4 and rel_2=-4 and rel_3=-4 and rel_4=-4 and rel_5=-4 and rel_6=-4 and rel_7=-4 and rel_8=-4 and
  rel_9=-4 and rel_10=-4 and rel_11=-4 and rel_12=-4 and rel_13=-4 and rel_14=-4 then do;
  indep=1;
end;
end;

**not legal;
do i=1 to 14;
  if (rel_a>28 and rel_a<37) then do; nlgrand=1; end;
  *spouse;          if rel_a=1 or rel_a=2 then do; nlsponse=1; end;
  *brother/sister;  if (rel_a>12 and rel_a<19) and age_a>20 then do; nlrelat=1; end;
  *aunt/uncle/other rel.; if (rel_a>69 and rel_a<85) and age_a>20 then do; nlrelat=1; end;
  *lover, roommate, other non-relative, mom's or dad's partner;
  if rel_a=69 or rel_a=68 or rel_a=85 or rel_a=88 or rel_a=89 then do; nlennrl=1; end;
  if rel_a=-1 or rel_a=-2 or rel_a=-3 then do; nlinv=1; end;
end;

/*problem is a variable to check whether R did not report a relationship code for one or more household members*/
problem=0;
do i=1 to 14; if rel_a=-1 or rel_a=-2 or rel_a=-3 then problem=1; end;

if age<18 then do;
  if nlgrand ne . and legal=0 then grand=nlgrand;          if nlsponse ne . and legal=0 then spouse=nlsponse;
  if nlrelat ne . and legal=0 then relat=nlrelat;          if nlennrl ne . and legal=0 then nonrel=nlennrl;
  if nlinv ne . and legal=0 then invalid=nlinv;
end;

*for all youths;
if nlsponse ne . then spouse=nlsponse;
do i=1 to 14;
  if rel_a=3 then momid=i;          if rel_a=4 then domid=i;
  if rel_a=5 then admom=i;         if rel_a=6 then adopdad=i;
  if rel_a=7 then stepma=i;        if rel_a=8 then stepda=i;
  if rel_a=9 then fostma=i;        if rel_a=10 then fostda=i;
end;

rel=-16;
if age>17 then do; rel=10; end;          if marry_1=-5 then do; rel=-5; end;
if indep=1 then do; rel=10; end;        if invalid=1 then do; rel=-3; end;
if nonrel>0 then do; rel=10; end;       if relat>0 then do; rel=9; end;
if grand>0 and momid=0 and domid=0 then do; rel=8; end;
if spouse=1 then do; rel=10; end;        if fostda>0 or fostma>0 then do; rel=7; end;
if admom>0 or adopdad>0 then do; rel=6; end;  if stepda>0 or stepma>0 then do; rel=10; end;
if domid>0 and momid=0 then do; rel=5; end;  if momid>0 and domid=0 then do; rel=4; end;
if domid>0 and momid=0 then do; if admom>0 or stepma>0 then rel=3; end;
if momid>0 and domid=0 then do; if adopdad>0 or stepda>0 then rel=2; end;
if momid>0 and domid>0 then do; both=1; rel=1; end;

if rel=-16 then rel=10;

endsas;
```

YOUTH'S MARITAL STATUS AND MARITAL/COHABITATION HISTORY

Variables Created:	CV_MARSTAT	CV_MARSTAT_COLLAPSED
	CV_FIRST_COHAB_DATE_M	CV_FIRST_COHAB_DATE_Y
	CV_FIRST_MARRY_DATE_M	CV_FIRST_MARRY_DATE_Y
	CV_FIRST_COHAB_MONTH	CV_FIRST_MARRY_MONTH
	CV_COHAB_TTL	CV_MARRIAGES_TTL

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
id	PUBID	m560022m, m560022y	YMAR-5600.02.02~M, ~Y
int_d, int_m, int_y	YINTDATE~D, ~M, ~Y	m570011m, m570011y	YMAR-5700.01.01~M, ~Y
m620	YMAR-620	m570021m, m570021y	YMAR-5700.02.01~M, ~Y
m650	YMAR-650	m610011, m610012	YMAR-6100.01.01, .02
m700-m718	YMAR-700-YMAR-718	m610021, m610022	YMAR-6100.02.01, .02
m729d	YMAR-729D	m63001-m63003	YMAR-6300.01-.03
m730	YMAR-730	m64001-m64003	YMAR-6400.01-.03
m740	YMAR-740	m700011m, m700011y	YMAR-7000.01.01~M, ~Y
m760	YMAR-760	m700012m, m700012y	YMAR-7000.01.02~M, ~Y
m1000	YMAR-1000	m700013m, m700013y	YMAR-7000.01.03~M, ~Y
m1500	YMAR-1500	m720011-m720013	YMAR-7200.01.01-.03
m2600	YMAR-2600	m727011-m727041	YMAR-7270.01.01-.04.01
m2800	YMAR-2800	m730011, m730021	YMAR-7300.01.01, .02.01
m30501-m30504	YMAR-3050.01-.04	m735011-m735041	YMAR-7350.01.01-.04.01
m31001m, m31001y	YMAR-3100.01~M, ~Y	m7900111, m7900211	YMAR-7900.01.01.01, .02.01.01
m31002m, m31002y	YMAR-3100.02~M, ~Y	m810111m, m810111y	YMAR-8100.01.01.01~M, ~Y
m31003m, m31003y	YMAR-3100.03~M, ~Y	m820111m, m820111y	YMAR-8200.01.01.01~M, ~Y
m31004m, m31004y	YMAR-3100.04~M, ~Y	m820211m, m820211y	YMAR-8200.02.01.01~M, ~Y
m45001-m45004	YMAR-4500.01-.04	m910011-m910013	YMAR-9100.01.01-.03
m46501-m46504	YMAR-4650.01-.04	m920011m, m920011y	YMAR-9200.01.01~M, ~Y
m46701	YMAR-4670.01	m920012m, m920012y	YMAR-9200.01.02~M, ~Y
m47001, m47004	YMAR-4700.01-.04	m920013m, m920013y	YMAR-9200.01.03~M, ~Y
m48001m, m48001y	YMAR-4800.01~M, ~Y	m9800111	YMAR-9800.01.01.01
m48002m, m48002y	YMAR-4800.02~M, ~Y	m101111m, m101111y	YMAR-10100.01.01.01~M, ~Y
m48003m, m48003y	YMAR-4800.03~M, ~Y	m1020111	YMAR-10200.01.01.01
m49001-m49003	YMAR-4900.01-.03	m1080011-m1080013	YMAR-10800.01.01-.03
m540011, m540012	YMAR-5400.01.01, .02	m114001-m114003	YMAR-11400.01-.03
m540021, m540022	YMAR-5400.02.01, .02	m117001m, m117001y	YMAR-11700.01~M, ~Y
m560011m, m560011y	YMAR-5600.01.01~M, ~Y	m117002m, m117002y	YMAR-11700.02~M, ~Y
m560012m, m560012y	YMAR-5600.01.02~M, ~Y		

Codes for Created Variable

Marital/Cohabitation Status

- | | |
|-----------------------------------|-------------------------------|
| 1 = never married, cohabiting | 6 = separated, not cohabiting |
| 2 = never married, not cohabiting | 7 = divorced, cohabiting |
| 3 = married, spouse present | 8 = divorced, not cohabiting |
| 4 = married, spouse absent | 9 = widowed, cohabiting |
| 5 = separated, cohabiting | 10 = widowed, not cohabiting |

Collapsed Marital Status

- 0 = never married
- 1 = married
- 2 = separated
- 3 = divorced
- 4 = widowed

This program creates two variables that describe marital status/cohabitation status as of the interview date for respondents age 16 and older. Other respondents are valid skips (-4). Note that later partners take precedence over earlier partners.

Appendix 3: Family Background and Formation Variable Creation

The program also creates variables that provide the dates of the youth's first marriage and/or cohabitation in both a continuous month scheme and as actual dates (for more information on the continuous month scheme, see appendix 7 in this document). Summary variables count the total number of marriages and cohabitations for each youth. Note that these variables are available only for youths age 16 and older as of 12/31/99. If a respondent is cohabiting and then marries it is considered both a cohabitation and a marriage. If someone refuses or doesn't know the full date of their marriage or cohabitation, then the spell is counted in the total variables and the date variables are coded -1 or -2 as applicable.

```
/**Create flag indicating whether r. said there was a
spouse in the house in the hh roster**/
spo_hous=0;
```

```
array hhrel (i) hhrel01-HHREL14;
do i=1 to 14;
  if hhrel=1 or hhrel=2 then spo_hous=1;
end;
```

```
*Create a variable indicating r.'s marital status as they
report it;
```

```
if m650 ge 0 then marques=m650;
else marques=m620;
```

```
/*Set up initial marital status using dli created variables. */
prevmsta=MARSTAT3;   prevcmst=CMARSTA3;
mar_ttl=mar_ttl3;    coh_ttl=coh_ttl3;
dlimary= FMAR_Yr3;   dlimarm= FMAR_Mr3;
dlimarc=fmarmr3;    dlicohm= FCOH_Mr3;
dlicohy= FCOH_Yr3;  dlicohcm=FCOHCMr3;
```

```
if cmarsta3=-5 then do; /*round 3 noninterview, go to
round 2*/
  prevmsta=marstat2;   prevcmst=cmarsta2;
  mar_ttl=mar_ttl2;   coh_ttl=coh_ttl2;
  dlimary= FMAR_Yr2;  dlimarm= FMAR_Mr2;
  dlimarc=fmarmr2;   dlicohm= FCOH_Mr2;
  dlicohy= FCOH_Yr2;  dlicohcm=FCOHCMr2;
end;
```

```
if cmarsta2=-5 and cmarsta3=-5 then do; /*round 3 and
round 2 noninterview, go to round 1*/
  prevmsta=marstat1;   prevcmst=cmarsta1;
  mar_ttl=mar_ttl1;   coh_ttl=coh_ttl1;
  dlimary= FMAR_Yr1;  dlimarm= FMAR_Mr1;
  dlimarc=fmarmr1;   dlicohm= FCOH_Mr1;
  dlicohy= FCOH_Yr1;  dlicohcm=FCOHCMr1;
end;
```

```
*Create interview date and previous interview date in
continuous months and months/years;
if int_y gt 0 and int_m gt 0 then do;
  doicm=(int_y-1980)*12+int_m;
end;
else if int_y eq -5 or int_m eq -5 then do;
  doicm=-5;
end;
```

```
else if int_y in(-1,-2, -3,-4) or int_m in(-1, -2, -3, -4)
  then do;
  doicm=-3;
end;
```

```
if int_yr3 gt 0 and int_mr3 gt 0 then do;
  dlicm=(int_yr3-1980)*12+int_mr3;
end;
else if int_yr3 = -5 and int_yr2>=-5 then do;
  dlicm=(int_yr2-1980)*12+int_mr2;
end;
else if int_yr2 = -5 and int_yr3 = -5 then do;
  dlicm=(int_yr1-1980)*12+int_mr1;
end;
else if int_yr3 in(-1,-2, -3,-4) or int_mr3 in(-1, -2, -3, -4)
  then do;
  dlicm=-3;
end;
else if int_yr2 in(-1,-2, -3,-4) or int_mr2 in(-1, -2, -3, -4)
  then do;
  dlicm=-3;
end;
```

```
/*set up arrays with marital/cohab status for each of the
continuous months*/
array m (l) m001-m260; /* married y/n */
array coh (l) coha001-coha260; /* cohab y/n */
array mars (l) mars001-mars260; /* marital status */
```

```
/* Initialize values to 0 for all who go through section,
between dli and current interview date. */
if m700>-4 then do; marstat=2; ttlm=0; ttlc=0; end;
```

```
do l=1 to 260;
  if dlicm le L le doicm then do; m=0; coh=0; mars=2;
  end;
end;
```

```
/* In the array section, an index by "p" indicates a loop
over partners, while an index by "t" indicates a loop
over time periods. */
```

```
/*year start cohabitating*/
array ysca (t) m31001y m700011y m700012y
m700013y;
array yscb (t) m31002y m700021y m700022y
m700023y;
array ysc (t) m31003y m700031y m700032y
m700033y;
```

Appendix 3: Family Background and Formation Variable Creation

```

array yscd (t) m31004y m700041y m700042y
    m700043y;

/*month start cohabitating*/
array msca (t) m31001m m700011m m700012m
    m700013m;
array msca (t) m31002m m700021m m700022m
    m700023m;
array msca (t) m31003m m700031m m700032m
    m700033m;
array msca (t) m31004m m700041m m700042m
    m700043m;

/*collapse all start years and months into arrays*/
array ysc (p) ysca yscb yscc yscd;
array msc (p) msca msca mscc msca;

/*legally married when began cohabitating*/
array mbega (t) m45001 m4500a2 m4500a3 m4500a4;
array mbega (t) m45002 m4500b2 m4500b3 m4500b4;
array mbega (t) m45003 m4500c2 m4500c3 m4500c4;
array mbega (t) m45004 m4500d2 m4500d3 m4500d4;

/*collapse married into array*/
array mbeg (p) mbega mbega mbega mbega;

/*dummy for cohabitation continuous - since dli*/
array cdli (t) m46701 cdli2 cdli3 cdli4;
array cdli (t) cdli1 cdli2 cdli3 cdli4;
array cdli (t) cdli1 cdli2 cdli3 cdli4;
array cdli (t) cdli1 cdli2 cdli3 cdli4;

/*dummy for cohabitation continuous - since start (after
dli date)*/
array cbega (t) m47001 m910011 m910012 m910013;
array cbega (t) m47002 cbega2 cbega3 cbega4;
array cbega (t) m47003 cbega2 cbega3 cbega4;
array cbega (t) m47004 cbega2 cbega3 cbega4;

/* continuous cohab vars into array*/
array cdli (p) cdli cdli cdli cdli;
array cbeg (p) cbega cbega cbega cbega;

/*year end cohabitating (not continuous)*/
array yeca (t) m48001y m920011y m920012y
    m920013y;
array yecb (t) m48002y yecb2 yecb3 yecb4;
array yecc (t) m48003y yecc2 yecc3 yecc4;
array yecd (t) m48004y yecd2 yecd3 yecd4;

/*month end cohabitating (not continuous)*/
array meca (t) m48001m m920011m m920012m
    m920013m;
array mecb (t) m48002m mecb2 mecb3 mecb4;
array mecc (t) m48003m mecc2 mecc3 mecc4;
array mecd (t) m48004m mecd2 mecd3 mecd4;

/*collapse all end years and months into arrays*/
array yec (p) yeca yecb yecc yecd;
array mec (p) meca mecb mecc mecd;

/* First or next change in status. Read "howa2" as change
in marital status for partner a, period 2. */
array howa1 (t) m540011 m540021 m9800111
    howa41;
array howa2 (t) m540012 m540022 howa32 howa42;
array howa3 (t) m7900111 howa32 howa33 howa43;
array howb3 (t) m7900211 howb32 howb33 howb43;

/*collapse into how status changed*/
array how1 (p) howa1 howb1 howc1 howd1;
array how2 (p) howa2 howb2 howc2 howd2;
array how3 (p) howa3 howb3 howc3 howd3;
array how4 (p) howa4 howb4 howc4 howd4;

/*month and year change is marry*/
array hyna1 (t) m570011y m101111y hyna13 hyna14;
array hyna1 (t) m570011m m101111m hyna23
    hyna24;
array hyna2 (t) m570012y hyna22 hyna23 hyna24;
array hyna2 (t) m570012m hyna22 hyna23
    hyna24;
array hyna3 (t) m820111m hyna32 hyna33
    hyna34;
array hyna3 (t) m820111y hyna32 hyna33 hyna34;
array hyna3 (t) m820211m hyna32 hyna33
    hyna34;
array hyna3 (t) m820211y hyna32 hyna33 hyna34;

/*collapse into month and year change is marry*/
array hym1 (p) hyna1 hyna1 hyna1 hyna1;
array hym1 (p) hyna1 hyna1 hyna1 hyna1;
array hym2 (p) hyna2 hyna2 hyna2 hyna2;
array hym2 (p) hyna2 hyna2 hyna2 hyna2;
array hym3 (p) hyna3 hyna3 hyna3 hyna3;
array hym3 (p) hyna3 hyna3 hyna3 hyna3;
array hym4 (p) hyna4 hyna4 hyna4 hyna4;
array hym4 (p) hyna4 hyna4 hyna4 hyna4;

/*month & year change=separation, divorce, annulment*/
array hysa1 (t) m560011y hysa12 hysa13 hysa14;
array hysa1 (t) m560011m hysa12 hysa13 hysa14;
array hysa2 (t) m560012y hysa22 hysa23 hysa24;
array hysa2 (t) m560012m hysa22 hysa23 hysa24;
array hysa3 (t) m810111y hysa32 hysa33 hysa34;
array hysa3 (t) m810111m hysa32 hysa33 hysa34;
array hysa4 (t) m810112y hysa42 hysa43 hysa44;
array hysa4 (t) m810112m hysa42 hysa43 hysa44;

/*collapse into month and year change=separation,
divorce, annulment*/
array hys1 (p) hysa1 hysb1 hysc1 hysd1;
array hys1 (p) hysa1 hysb1 hysc1 hysd1;
array hys2 (p) hysa2 hysb2 hysc2 hysd2;

```

Appendix 3: Family Background and Formation Variable Creation

```

array hms2 (p) hmsa2 hmsb2 hmhc2 hmsd2;
array hys3 (p) hysa3 hysb3 hysc3 hysd3;
array hms3 (p) hmsa3 hmsb3 hmhc3 hmsd3;
array hys4 (p) hysa4 hysb4 hysc4 hysd4;
array hms4 (p) hmsa4 hmsb4 hmhc4 hmsd4;

/* marital status at end of cohab period */
array estata (t) m63001 m1080011m m1080012m
m1080013m;
array estatb (t) m63002 estatb2 estatb3 estatb4;
array estatc (t) m63003 estatc2 estatc3 estatc4;
array estasd (t) m63004 estasd2 estasd3 estasd4;

/* collapse marital status into one array */
array estat (p) estata estatb estatc estasd;

/* month and year relationship ended */
array eyra (t) m117001y eyra2 eyra3 eyra4;
array eyrb (t) m117002y eyrb2 eyrb3 eyrb4;
array eyrc (t) eyrc1 eyrc2 eyrc3 eyrc4;
array eyrd (t) eyrd1 eyrd2 eyrd3 eyrd4;
array emoa (t) m117001m emoa2 emoa3 emoa4;
array emob (t) m117002m emob2 emob3 emob4;
array emoc (t) emoc1 emoc2 emoc3 emoc4;
array emod (t) emod1 emod2 emod3 emod4;

/*collapse all start years and months into arrays*/
array eyr (p) eyra eyrb eyrc eyrd;
array emo (p) emoa emob emoc emod;

/* current status of relationship */
array fstata (t) m114001 fstata2 fstata3 fstata4;
array fstatb (t) m114002 fstatb2 fstatb3 fstatb4;
array fstatc (t) m114003 fstatc2 fstatc3 fstatc4;
array fstatd (t) m114004 fstatd2 fstatd3 fstatd4;

/*collapse married into array*/
array fstat (p) fstata fstatb fstatc fstatd;

/* Widow flag */
array widowa (t) widowa1 widowa2 widowa3 widowa4;
array widowb (t) widowb1 widowb2 widowb3 widowb4;
array widowc (t) widowc1 widowc2 widowc3 widowc4;
array widowd (t) widowd1 widowd2 widowd3
widowd4;
array widow (p) widowa widowb widowc widowd;

/*continuous start month of cohabitation*/
array csmca (t) csmca1 csmca2 csmca3 csmca4;
array csmcb (t) csmcb1 csmcb2 csmcb3 csmcb4;
array csmcc (t) csmcc1 csmcc2 csmcc3 csmcc4;
array csmcd (t) csmcd1 csmcd2 csmcd3 csmcd4;
array csmc (p) csmca csmcb csmcc csmcd;

/*continuous end month of cohabitation*/
array cemca (t) cemca1 cemca2 cemca3 cemca4;
array cemcb (t) cemcb1 cemcb2 cemcb3 cemcb4;

array cemcc (t) cemcc1 cemcc2 cemcc3 cemcc4;
array cemcd (t) cemcd1 cemcd2 cemcd3 cemcd4;
array cemc (p) cemca cemcb cemcc cemcd;

/*continuous start month of marriage*/
array csmma (t) csmma1 csmma2 csmma3 csmma4;
array csmmb (t) csmmb1 csmmb2 csmmb3 csmmb4;
array csmmc (t) csmmc1 csmmc2 csmmc3 csmmc4;
array csmmd (t) csmmd1 csmmd2 csmmd3 csmmd4;
array csmm (p) csmma csmmb csmmc csmmd;

/*continuous start month of separation, etc*/
array csmsa (t) csmsa1 csmsa2 csmsa3 csmsa4;
array csmsb (t) csmsb1 csmsb2 csmsb3 csmsb4;
array csmsc (t) csmsc1 csmsc2 csmsc3 csmsc4;
array csmsd (t) csmsd1 csmsd2 csmsd3 csmsd4;
array csms (p) csmsa csmsb csmsc csmsd;

/* married at date of last interview */
array mdlia (t) mdlia1 mdlia2 mdlia3 mdlia4;
array mdlib (t) mdlib1 mdlib2 mdlib3 mdlib4;
array mdlic (t) mdlic1 mdlic2 mdlic3 mdlic4;
array mdlid (t) mdlid1 mdlid2 mdlid3 mdlid4;
array mdli (p) mdlia mdlib mdlic mdlid;

/*Collapsed marital status*/
array cmarsa (t) cmarsa1 cmarsa2 cmarsa3 cmarsa4;
array cmarsb (t) cmarsb1 cmarsb2 cmarsb3 cmarsb4;
array cmarsc (t) cmarsc1 cmarsc2 cmarsc3 cmarsc4;
array cmarsd (t) cmarsd1 cmarsd2 cmarsd3 cmarsd4;
array cmars (p) cmarsa cmarsb cmarsc cmarsd;

/*number of cohabitations*/
array nuca (t) nuca1 nuca2 nuca3 nuca4;
array nucb (t) nucb1 nucb2 nucb3 nucb4;
array nucc (t) nucc1 nucc2 nucc3 nucc4;
array nucd (t) nucd1 nucd2 nucd3 nucd4;
array nuc (p) nuca nucb nucc nucd;

/*number of marriages*/
array numa (t) numa1 numa2 numa3 numa4;
array numb (t) numb1 numb2 numb3 numb4;
array numc (t) numc1 numc2 numc3 numc4;
array numd (t) numd1 numd2 numd3 numd4;
array num (p) numa numb numc numd;

/*dummy set to one if start month cohabitate is invalid*/
array fixmc1 (t) fixmc11 fixmc12 fixmc13 fixmc14;
array fixmc2 (t) fixmc21 fixmc22 fixmc23 fixmc24;
array fixmc3 (t) fixmc31 fixmc32 fixmc33 fixmc34;
array fixmc4 (t) fixmc41 fixmc42 fixmc43 fixmc44;
array fixmc (p) fixmc1 fixmc2 fixmc3 fixmc4;

/*dummy set to one if start month marry is invalid*/
array fixmm1 (t) fixmm11 fixmm12 fixmm13 fixmm14;
array fixmm2 (t) fixmm21 fixmm22 fixmm23 fixmm24;
array fixmm3 (t) fixmm31 fixmm32 fixmm33 fixmm34;

```

Appendix 3: Family Background and Formation Variable Creation

```

array fixmm4 (t) fixmm41 fixmm42 fixmm43 fixmm44;
array fixmm (p) fixmm1 fixmm2 fixmm3 fixmm4;

/*dummy set to one if start year cohabitate is invalid*/
array fixyc1 (t) fixyc11 fixyc12 fixyc13 fixyc14;
array fixyc2 (t) fixyc21 fixyc22 fixyc23 fixyc24;
array fixyc3 (t) fixyc31 fixyc32 fixyc33 fixyc34;
array fixyc4 (t) fixyc41 fixyc42 fixyc43 fixyc44;
array fixyc (p) fixyc1 fixyc2 fixyc3 fixyc4;

/*dummy set to one if start year marry is invalid*/
array fixym1 (t) fixym11 fixym12 fixym13 fixym14;
array fixym2 (t) fixym21 fixym22 fixym23 fixym24;
array fixym3 (t) fixym31 fixym32 fixym33 fixym34;
array fixym4 (t) fixym41 fixym42 fixym43 fixym44;
array fixym (p) fixym1 fixym2 fixym3 fixym4;

/* This flag prevents the dli from being imputed as the c.v
for marriage month. Since the program assigns the
dli to the start date of marriage, this is not what
should go into the c.v. */
array flagcm1 (t) flagcm11 flagcm12 flagcm13 flagcm14;
array flagcm2 (t) flagcm21 flagcm22 flagcm23 flagcm24;
array flagcm3 (t) flagcm31 flagcm32 flagcm33 flagcm34;
array flagcm4 (t) flagcm41 flagcm42 flagcm43 flagcm44;
array flagcm (p) flagcm1 flagcm2 flagcm3 flagcm4;

/* Is this continuous end month of spearation? Is it not
found anywhere else in the program. */
array cems (p) cems1 cems2 cems3 cems4;

/* Begin by coding respondents who have been
cohabitating or married since dli. Set begin date of
cohab. (or marriage) as dli. */

/* For respondents who report no beginning date of
cohab. because they were cohabitating at dli, the
start date of cohab must be created for the program.
This is done outside the loop because the dli partner
must be partner "a" and begin at loop 1. For the
program, the start date of cohab is set to dli. The
same is done for marriages. */
if m714=1 and m30501=1 then csmca1=dlicm;
if prevcmst=1 and m714=1 and m30501=1 then do;
csmma1=dlicm;
mdlia1=1;
cmarsa1=1;
flagcma1=1;
end;

/* These respondents have gotten married and during
Round 4 but do not have a start month of
separation because of the strange way they go
through the loop section. */
if id in (1309501, 1550021, 1782191, 1787411,
1928761, 1932841) then do;
csmsa1=doicm; handedit=1;
end;

end;

do p=1 to 4;
do t=1 to 4;

if cdli=1 then do;
csmc=dlicm;
flagcm=1;
end;

/* Code respondents divorced at dli, using dli c.v. Set
begin date of separation as dli. */
if mdli=3 then do;
cmars=3;
csms=dlicm;
flagcm=1;
end;

/* If married at dli is don't know or refuse, then set
collapsed marital status to -1 or -2. */
if mdli in (-1,-2) then cmars=mdli;

/* Create continuous start month cohab variable from
data, create dummies for negative answers. Any
negative answer is set to the dli and dummies (fixyc,
fixmc) are created to identify these cases. */
if msc in (-1,-2,-3) then do;
fixmc=1;
csmomiss=1;
end;

/* Hand edit bad start month to January, if valid start
year is given. */
if msc in (-1,-2,-3) and ysc>0 then do;
msc=1;
end;

if ysc>0 and msc>0 then csmc=(ysc-1980)*12 + msc;

/* If a start month is missing, set the start month to Jan.
(coded above) If a start year is missing, set to dli. */
if ysc in (-1,-2,-3) then do;
csmc=dlicm; fixyc=1; csyrmiss=1;
end;

/* Create continuous end month cohab variable from
data. If continuous end month of cohabitation is
missing, not cohabiting at dli (cdli ne 1) and
cohabiting continuously with new partner (cbeg ne
0), then ymec=int date. */
/* Hand edit bad end month to Dec, if valid start year
is given. */
if mec in (-1,-2,-3) and yec>0 then do; mec=12; end;
if yec>0 and mec>0 then cemc=(yec-1980)*12 + mec;
if yec in (-1,-2,-3) then do; cemc=doicm; ceyrmiss=1;
end;
if mec in (-1,-2,-3) then do; cemomiss=1; end;

```

Appendix 3: Family Background and Formation Variable Creation

```

/* Set upper bound on imputed month values. */
if mec in (-1,-2,-3) and cemc>doicm then do;
    cemc=doicm; end;

/* If cohab since dli or married at beginning of cohab,
then set cohab end date as int date. */
if cbeg=1 or cdli=1 then cemc=doicm;

/* If married when began cohab, then assume married and
change date arrays and update marriage date flags. */
if mbeg=1 or mdli=1 then do;
    cmars=1;
    csmm=csmc;
    if fixmc=1 then fixmm=1;
    if csmomiss=1 then msmomiss=1;
    if fixyc=1 then fixym=1;
    if cysrmiss=1 then msyrmiss=1;

/* Set flagcm back to dot, since we now have a date
for marriage, instead of dli. */
flagcm=.;

/*set separation date as the end month of cohab*/
if cemc>0 then csms=cemc;

/* Marriage ends in legal separation */
if how1=3 then cmars=2; if how2=3 then cmars=2;
if how3=3 then cmars=2; if how4=3 then cmars=2;

/* Marriage ends in divorce */
if how1=4 then cmars=3; if how2=4 then cmars=3;
if how3=4 then cmars=3; if how4=4 then cmars=3;

/* Marriage ends in divorce */
if how1=5 then cmars=0; if how2=5 then cmars=0;
if how3=5 then cmars=0; if how4=5 then cmars=0;
end;

/* For respondentds whose marital status changes since
dli, update marriage/separation/cohab dates and
collapsed marital status. */
/* Change is marriage: beginning of loop. */
if how1=1 and hym1>0 and hmm1>0 then do;
    cmars=1; csmm=(hym1-1980)*12 + hmm1;
end;
if how1=1 and (-4<hym1<0 or -4<hmm1<0) then do;
    cmars=1; csmm=doicm;
end;
/* Change is legal separation. */
if how1=3 and hys1>0 and hms1>0 then do;
    cmars=2; csms=(hys1-1980)*12 + hms1;
end;
if how1=3 and (-4<hym1<0 or -4<hmm1<0) then do;
    cmars=2; csms=doicm; meyrmiss=1;
end;
/* Change is divorce. */
if how1=4 and hys1>0 and hms1>0 then do;
    cmars=3; csms=(hys1-1980)*12 + hms1;
end;
if how1=4 and (-4<hym1<0 or -4<hmm1<0) then do;
    cmars=3; csms=doicm; meyrmiss=1;
end;
/* Change is annulment. */
if how1=5 and hys1>0 and hms1>0 then do;
    cmars=0; csms=(hys1-1980)*12 + hms1;
end;
if how1=5 and (-4<hym1<0 or -4<hmm1<0) then do;
    cmars=0; csms=doicm; meyrmiss=1;
end;

if how1 in (-1,-2) then cmars=how1;

/*At this point the program repeats the above code (from
"beginning of loop" for marriages 2-4 (how2-how4).
Contact NLS User Services for more information.*/

/* This loop collects information from the marital status
at the end of relationship period. Code: 0 = R and
partner not married, 1 = R and partner married, 2 =
R and partner separated, 3 = R and partner
divorced, 4 = R widowed. */
if estat>-4 then cmars=estat;

/* YMAR-114000 (fstat in the arrays) collects current
status of the relationship. */
/* fstat=annulment, which translates into never married in
collapsed mar. status */
if fstat=2 and eyr>0 and emo>0 then do;
    cmars=0; csms=(eyr-1980)*12 + emo;
end;
if fstat=2 and (-4<eyr<0 or -4<emo<0) then do;
    cmars=0; csms=doicm;
end;
/* separation */
if fstat=3 and eyr>0 and emo>0 then do;
    cmars=2; csms=(eyr-1980)*12 + emo;
end;
if fstat=3 and (-4<eyr<0 or -4<emo<0) then do;
    cmars=2; csms=doicm;
end;
/* divorce */
if fstat=4 and eyr>0 and emo>0 then do;
    cmars=3; csms=(eyr-1980)*12 + emo;
end;
if fstat=4 and (-4<eyr<0 or -4<emo<0) then do;
    cmars=3; csms=doicm;
end;
/* widowed */
if fstat=5 and eyr>0 and emo>0 and cmars=1 then do;
    cmars=4; csms=(eyr-1980)*12 + emo; cemc=(eyr-
1980)*12 + emo; widow=1;
end;

```

Appendix 3: Family Background and Formation Variable Creation

```

if fstat=5 and (-4<eyr<0 or -4<emo<0) and cmars=1
  then do;
    cmars=4;          csms=doicm;
    cemc=doicm;      meyrmiss=1;
    ceyrmiss=1;      widow=1;
  end;
if -4<fstat<0 then cmars=fstat;

/* If there is no change in marital status since dli or
beginning of marriage and the respondent reports
being married at dli (cmars=1), then let the time of
separation be the interview date. */
if cmars=1 and (cdli=1 or cbeg=1) then csms=doicm;
if csms>0 then cems=doicm;
if csmm>0 then cemm=doicm;

/* Count the total number of cohabs. */
if csmc>0 then do;
  tlc=tlc+1; nuc=tlc;
  if nuc=1 then r4cohcm=csmc;
end;
/* Count the total number of marriages */
if csmm>0 then do;
  tlm=tlm+1; num=tlm;
  if num=1 then r4marcm=csmm;
end;
if flagcm=1 then dlist=1;
corrc=0;
corrm=0;

/* Create a correction variable to prevent double counting
the same marriage/cohab partner since we set the
start months of marriage/cohab at dli if the person
ha a partner at dli. */
if tlc>-1 or tlm>-1 then do;
  if csmca2>0 and csmca2 ne . then corrc=corrc+1;
  if csmcb2>0 and csmcb2 ne . then corrc=corrc+1;
  if csmcc2>0 and csmcc2 ne . then corrc=corrc+1;
  if csmcd2>0 and csmcd2 ne . then corrc=corrc+1;
  if csmma2>0 and csmma2 ne . then corrm=corrm+1;
  if csmbb2>0 and csmbb2 ne . then corrm=corrm+1;
  if csmmc2>0 and csmmc2 ne . then corrm=corrm+1;
  if csmd2>0 and csmd2 ne . then corrm=corrm+1;
  if csmca3>0 and csmca3 ne . then corrc=corrc+1;
  if csmcb3>0 and csmcb3 ne . then corrc=corrc+1;
  if csmcc3>0 and csmcc3 ne . then corrc=corrc+1;
  if csmcd3>0 and csmcd3 ne . then corrc=corrc+1;
  if csmma3>0 and csmma3 ne . then corrm=corrm+1;
  if csmbb3>0 and csmbb3 ne . then corrm=corrm+1;
  if csmmc3>0 and csmmc3 ne . then corrm=corrm+1;
  if csmd3>0 and csmd3 ne . then corrm=corrm+1;
  if csmca4>0 and csmca4 ne . then corrc=corrc+1;
  if csmcb4>0 and csmcb4 ne . then corrc=corrc+1;
  if csmcc4>0 and csmcc4 ne . then corrc=corrc+1;
  if csmcd4>0 and csmcd4 ne . then corrc=corrc+1;
  if csmma4>0 and csmma4 ne . then corrm=corrm+1;
  if csmbb4>0 and csmbb4 ne . then corrm=corrm+1;

  if csmmc4>0 and csmmc4 ne . then corrm=corrm+1;
  if csmd4>0 and csmd4 ne . then corrm=corrm+1;
end;

/* Subtract the correction from the count variable to
create the total number of marriages & total number
of cohabs c.v. */
tlcnew=tlc-corr;
tlmnew=tlm-corr;

/*Create check variable for missing start/stop months of
marriage and cohabitation. */
if csmc>. and cemc=. then cerror=1;
if csmc=. and cemc>. then cerror=1;
if csmm>. and csms=. then merror=1;
if csmm=. and csms>. then merror=1;

/*correct for rs who don't know start month of
cohabiting*/
if fixmc=1 and (msc<0 and ysc>-4) then do;
  r4cohy=ysc; r4cohcm=-3; r4cohmm=-3;
end;

/*correct for rs who don't know start year of cohabiting*/
if fixyc=1 and (ysc<0 and msc>-4) then do;
  r4cohm=msc; r4cohcm=-3; r4cohy=-3;
end;

/* correct for rs who don't know start year and month of
cohabiting*/
if fixyc=1 and fixmc=1 then do;
  r4cohm=-3; r4cohcm=-3; r4cohy=-3;
end;

/*correct for rs who don't know start month of marriage*/
if fixmm=1 and (ysc<0 and msc>-4) then do;
  r4mary=ysc; r4marcm=-3; r4marm=-3;
end;

/*correct for rs who don't know start year of marriage*/
if fixym=1 and (ysc<0 and msc>-4) then do;
  r4marm=msc;
  r4marcm=-3;
  r4mary=-3;
end;

/*correct for rs who don't know start year and month of
marriage*/
if fixym=1 and fixmm=1 then do;
  r4marm=-3; r4marcm=-3; r4mary=-3;
end;

/* Now that the start & end dates of marriages/cohabs are
created, we place that info into the array. The info
from the last week is used to create the variable for
marital status and collapsed marital status. */
C=0;

```

Appendix 3: Family Background and Formation Variable Creation

```

do L=1 to 260;
  C=C+1;
  if csmc>0 and cemc>0 and csmc <= C <= cemc then
    coh=1;
  if cmars=1 and csmm>0 and csms>0 and csmm le c le
    csms then m=1;
  if cmars in (2,3) and csms>0 and cems>0 and csms le
    c le cems then m=cmars;
  if cmars in (-2,-3) then m=cmars;
  if c=doicm then do;
    if m in (-1,-2) then mars=m;
    if m=0 and coh=1 then mars=1;
    if m=0 and coh=0 then mars=2;
    if m=1 and coh=1 then mars=3;
    if m=1 and coh=0 then mars=4;
    if m=2 and coh=1 then mars=5;
    if m=2 and coh=0 then mars=6;
    if m=3 and coh=1 then mars=7;
    if m=3 and coh=0 then mars=8;
    if m=4 and coh=1 then mars=9;
    if m=4 and coh=0 then mars=10;
    marstat=mars;
  end;
end;
end;
end;

/*Create collapsed marital status from marital status.*/
if 1 le marstat le 2 then cmarstat=0;
if 3 le marstat le 4 then cmarstat=1;
if 5 le marstat le 6 then cmarstat=2;
if 7 le marstat le 8 then cmarstat=3;
if 9 le marstat le 10 then cmarstat=4;
if -2 le marstat le -1 then cmarstat=marstat;

/* For 10 respondents, their Round 3 c.v. for first
  marriage date post-dates a marriage they report in
  Round 4. We accept the Round 3 data as true, since
  it is closer to the time of marriage/cohabitation.*/
if dlimarcm>0 then marcm=dlimarcm;
if r4marcm>0 then marcm=r4marcm;
if dlimarcm>0 and r4marcm>0 then marcm=dlimarcm;
if dlcohcsm>0 then cohcm=dlcohcsm;
if r4cohcsm>0 then cohcm=r4cohcsm;
if dlcohcsm>0 and r4cohcsm>0 then cohcm=dlcohcsm;
if -4<r4marcm<0 and dlimarcm=-4 then
  marcm=r4marcm;
if r4marcm=-4 and 0<dlimarcm<-4 then
  marcm=dlimarcm;
if -4<r4cohcsm<0 and dlcohcsm=-4 then cohcm=r4cohcsm;
if r4cohcsm=-4 and 0<dlcohcsm<-4 then
  cohcm=dlcohcsm;

/* Using continuous month information, create month
  and year of marriage and cohabitation.*/
array cvm cohcm marcm;
array cvy coh y mary;

array cvm cohcm marcm;
do over cvcm;
  if 253 le cvcm le 264 then do; cvy=2001; cvm=cvcm-
    252; end;
  if 241 le cvcm le 252 then do; cvy=2000; cvm=cvcm-
    240; end;
  if 229 le cvcm le 240 then do; cvy=1999; cvm=cvcm-
    228; end;
  if 217 le cvcm le 228 then do; cvy=1998; cvm=cvcm-
    216; end;
  if 205 le cvcm le 216 then do; cvy=1997; cvm=cvcm-
    204; end;
  if 193 le cvcm le 204 then do; cvy=1996; cvm=cvcm-
    192; end;
  if 181 le cvcm le 192 then do; cvy=1995; cvm=cvcm-
    180; end;
  if 169 le cvcm le 180 then do; cvy=1994; cvm=cvcm-
    168; end;
  if 157 le cvcm le 168 then do; cvy=1993; cvm=cvcm-
    156; end;
  if cvcm=. then do; cvcm=-4; cvy=-4; cvm=-4; end;
  if cvcm=-3 then do; cvcm=-3; cvy=-3; cvm=-3; end;
end;

/*if old date present and don't deny then use old date*/
if dlcohcsm ne -4 and (m712=1 or m714=1 or m716=0 or
  (m712=0 and m718=1)) then do; cohcm=dlcohcsm;
  coh=dlcohy; cohcm=dlcohcsm; end;
if dlimarm gt -4 and (m712=1 or m714=1 or m716=0)
  then do; marcm=dlimarcm; mary=dlimary;
  marm=dlimarm; end;

/*previously in relationship, in more since interview*/
/* These respondents had a relationship prior to dli, then
  a new relationship during the R4 interview period.
  Here we simply add the previous c.v. to the Round
  4 count of cohab/marriages for the total c.v.*/
if m712=1 and m740=1 and ttlmnew=>0 and mar_ttl=>0
  then do; ttlmnew=tlmnew+mar_ttl; end;
if m712=1 and m740=1 and ttlcnew=>0 and coh_ttl=>0
  then do; ttlcnew=tlcnew+coh_ttl; end;

/*in relationship at int date, no more since interview*/
/* These respondents had a relationship at dli and no
  other relationships since. This includes people who
  lived and/or married to the same person throughout
  and people whose relationship ended in the R4
  interview period. By definition, their total cohab.
  and total marriage c.v. should be the same as their
  dli c.v.'s.*/
if m714=1 and m730=0 and coh_ttl>-4 then do;
  ttlcnew=coh_ttl; end;
if m714=1 and m730=0 and mar_ttl>-4 then do;
  ttlmnew=mar_ttl; end;

/*in relationship at int date, in more since interview*/

```

Appendix 3: Family Background and Formation Variable Creation

```

/* These respondents have relationships at dli, and have
   other relationships during the interview period. The
   counter variable setup corrc will erase one of these
   relationships, but this is the one case where it
   shouldn't. There are 6 of these people. */
if m714=1 and m730=1 and ttlmnew=>0 and mar_ttl=>0
then do; ttlmnew=ttlmnew+mar_ttl; end;
if m714=1 and m730=1 and ttlcnew=>0 and coh_ttl=>0
then do; ttlcnew=ttlcnew+coh_ttl; end;

***Change marstat if r. said s/he was married and said
there was a spouse in hh, but misunderstood later
questions about changes in marital status;
/* These respondents indicate they are married (YMAR-
620) and have a spouse present in the HH (see HH
roster), but give no date for marriage. There is a
chance the skip pattern confuses the respondent,
thinking they already told us about the marriage at
YMAR-620 and then saying there are no changes in
the marital status, which is where the marriage date
would come from. The first marriage date c.v.'s are
set to -3, the marital status (collapsed and regular)
is changed to show marriage, the total number of
marriages is set to -3, and the cohab dates and
counter are unchanged. The total number of
marriages is carried over from the previous round. */
***For pubid=3955, log said partner was away in
treatment so she really was married;
**Exclude pubid=8545 because log says not legally
married;
fixflag=0;
if cmarstat eq 0 and m620=1 and spo_hous = 1 /* and
pubid ne 8545 */ then do;
  cmarstat=1;
  if marstat=1 then marstat=3;
  if marstat=2 then marstat=4;
  fixflag=1;          ttlmnew=mar_ttl;
  ttlcnew=coh_ttl;   marcm=dlimarcm;

  mary=dlimary;      marm=dlimarm;
end;

/* The respondents who refuse questions throughout the
marriage section are coded as -3 for marital status,
collapsed marital status, and total number of
marriages/cohabs. First date of marriage/cohab is
whatever was in the previous round. */
if m45001=-1 or m45002=-1 or M7900111 in (-1,-2)
then do; marstat=-3; cmarstat=-3; ttlmnew=-3; end;
if m740 in (-1,-2) then do; marstat=-3; cmarstat=-3;
  ttlmnew=-3; ttlcnew=-3; end;

/* The remaining merror people (9 of them - there are no
error) have a start date of separation but no start
date of marriage. since it is difficult to tell whether
these people were married to begin with, we are
coding them as -3. */
if merror=1 or error=1 then do; marstat=-3; cmarstat=-3;
  ttlmnew=-3; end;
if m700=-4 then do; marstat=-4; cmarstat=-4;
  ttlmnew=-4; ttlcnew=-4; cohcm=-4;
  marcm=-4;cohy=-4; mary=-4; cohmm=-4; marm=-4;
end;
if m700=-5 then do; marstat=-5; cmarstat=-5;
  ttlmnew=-5; ttlcnew=-5; cohcm=-5; marcm=-5;
  cohy=-5; mary=-5; cohmm=-5; marm=-5; end;
if marstat in (-1,-2) then marstat=-3;
if cmarstat in (-1,-2) then cmarstat=-3;
if cohcm in (-1,-2) then cohcm=-3;
if marcm in (-1,-2) then marcm=-3;
if cohmm in (-1,-2) then cohmm=-3;
if cohy in (-1,-2) then cohy=-3;
if marm in (-1,-2) then marm=-3;
if mary in (-1,-2) then mary=-3;

endsas;

```

YOUTH'S FERTILITY AND CHILD STATUS

Variables Created: CV_CHILD_BIRTH_DATE.xx_M CV_CHILD_BIRTH_DATE.xx_Y
 CV_CHILD_DEATH_DATE.xx_M CV_CHILD_DEATH_DATE.xx_Y
 CV_CHILD_BIRTH_MONTH.xx CV_CHILD_DEATH_MONTH.xx
 CV_CHILD_STATUS.xx CV_BIO_CHILD_NR
 CV_BIO_CHILD_HH

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
oldsta01- oldsta04	YFER-1850.01-YFER-1850.04	bm03, by03	BIOCHILD_BDATE.03~M, ~Y
dmold, dyold	YFER-1860.01~M, ~Y	bioid01- bioid03	BIOCHILD_ID.01- BIOCHILD_ID.03
bmnew01, bynew01	YFER-5600.01~M, ~Y	biouid01- biouid03	BIOCHILD_UID.01- BIOCHILD_UID.03
bmnew02, bynew02	YFER-5600.02~M, ~Y	id	PUBID
newsta01, newsta02	YFER-5900.01, .02	bmold01, byold01	TOTBIOCHILDUPD2_BDATE.01~M, ~Y
dmnew, dynew	YFER-6000.01~M, ~Y	bmold02, byold02	TOTBIOCHILDUPD2_BDATE.02~M, ~Y
bm01, by01	BIOCHILD_BDATE.01~M, ~Y	bmold03, byold03	TOTBIOCHILDUPD2_BDATE.03~M, ~Y
bm02, by02	BIOCHILD_BDATE.02~M, ~Y	pubid	PUBID

Codes for Created Variables

Date of birth and death variables
 Date variables are presented as both the actual month and year and the month number in a continuous month scheme.

Status variables
 1 Adopted
 2 Deceased
 3 Non-resident, foster care
 4 Non-resident, not adopted or in foster care
 5 Resident

This program creates a number of variables describing the youth's fertility and the current status of the youth's children. For more information on the continuous month system, see appendix 7 in this document.

```
/* first, create a variable indicating the bmonth(i), birth
    month, and byear(i), birth year, for each biological
    child. Use variables from roster: BIOCHILD, which
    is the "clean" roster for all bio children */
array bm(3) bm01 bm02 bm03;
array by(3) by01 by02 by03;
array bmonth(3) bmonth01 bmonth02 bmonth03;
array byear(3) byear01 byear02 byear03;

/*small problem with recoding*/
if id=1735931 and by01=-2 then do;
    bm01=-3; by01=-3;
end;
/*correct the bdate when report a differing date of birth
    and age (HHIFINAL) in r4 (based on R5 data)*/
if id=1353832 then do;
    bm01=12; by01=1999; bmold01=12; byold01=1999;
end;
if id=1139502 then do;
    bm02=-3; by02=-3; bmold02=-3; byold02=-3;
    oldsta02=1;
end;
if id=1639431 then do;
    bm02=-4; by02=-4;
end;
```

```
/*don't know the bdate when report an early date of birth
    (pre-1990) and a differing age (HHIFINAL) in r4*/
if id=1647561 then do;
    bm01=-3; by01=-3; bmold01=-3; byold01=-3;
end;
/*correct the YFER report conflicts with BIOCHILD
    report*/
if id=1223531 then do;
    bm02=9; by02=2000;
end;
if id=1722971 then do;
    bmold02=6; byold02=1998; oldsta02=1;
end;
/*problem date resolved in r4 comments*/
if id=1830422 then byold02=2000;

do i=1 to 3; bmonth(i)=bm(i); byear(i)=by(i);
end;

/* second, create a continuous month scheme variable for
    the month of birth of the children using the
    formula: (12*(byear(i)-1980)+bmonth(i)) */
array mob(3) mob01 mob02 mob03;
do i=1 to 3;
    mob(i)=-4;
    if bmonth(i) eq -3 or byear(i) eq -3 then mob(i)=-3;
```

Appendix 3: Family Background and Formation Variable Creation

```

if bmonth(i) gt 0 and byear(i) gt 0 then
  mob(i)=12*(byear(i)-1980)+bmonth(i);
if bmonth(i) eq -5 then mob(i)=-5;
end;

/* third, create an actual date variable for the date of
death of the youth's children. For old reported
babies, use TOTBIOCHILDUPD2 roster birthday
and match up with birthday in BIOCHI roster,
yfer1860 variables are used to get the date of death.
For new reported babies, use new baby birthday
from the fertility section(yfer5600) and match up
with birthday in BIOCHI roster, yfer6000 variables
are used to get the date of death. */
array bmnew(3) bmnew01 bmnew02 bmnew03;
array bynew(3) bynew01 bynew02 bynew03;
array bmold(3) bmold01 bmold02 bmold03;
array byold(3) byold01 byold02 byold03;
array bioid(3) bioid01 bioid02 bioid03;
array dmonth(3) dmonth01 dmonth02 dmonth03;
array dyear(3) dyear01 dyear02 dyear03;
array newsta(3) newsta01 newsta02 newsta03;
array oldsta(3) oldsta01 oldsta02 oldsta03;

do i=1 to 3;
  dmonth(i)=dmold; dyear(i)=dyold;
  if bmonth(i)=-5 then dmonth(i)=-5;
  if byear(i)=-5 then dyear(i)=-5;
end;

do i=1 to 3;
  if (bm(i)=bmold01 and by(i)=byold01) then do;
    dmonth(i)=dmold; dyear(i)=dyold;
  end;
  if (bm(i)=bmnew01 and by(i)=bynew01) then do;
    dmonth(i)=dmnew; dyear(i)=dynew;
  end;
end;

/* fourth, create a continuous month scheme variable for
the month of death of the children using the
formula: (12*(dody(i)-1980)+dodm(i)) */
array mod(3) mod1 mod2 mod3;

do i=1 to 3;
  if dmonth(i)=-4 then mod(i)=-4;
  if dmonth(i)=-3 then mod(i)=-3;
  if dmonth(i) gt 0 and dyear(i) ge 1980 then
    mod(i)=12*(dyear(i)-1980)+dmonth(i);
  if dmonth(i)=-5 then mod(i)=-5;
end;

/* fifth, create a variable indicating the status of youth's
first (second, third) child:
The old reported baby's status is collected through
yfer1850, and the new reported baby's status is
collected through yfer5900 */
array status(3) status01 status02 status03;

/*no match on bmold and byold for some r's due to
coding problem & changed here*/
if id=1735931 or id=1032671 then do;
  bmold01=-3; byold01=-3;
end;

/* coding status according to the new questionnaire */
do i=1 to 3;
  if (bm(i)=bmold01 and by(i)=byold01) then do;
    status(i)=oldsta01;
  end;
  if (bm(i)=bmold02 and by(i)=byold02) then do;
    status(i)=oldsta02;
  end;
  if (bm(i)=bmold03 and by(i)=byold03) then do;
    status(i)=oldsta03;
  end;
end;

do i=1 to 3;
  if (bm(i)=bmnew01 and by(i)=bynew01) then do;
    status(i)=newsta01;
  end;
  if (bm(i)=bmnew02 and by(i)=bynew02) then do;
    status(i)=newsta02;
  end;
end;

/* for kids who don't have a birthdate on the
totbioupdate roster, this should not happen in
round 5*/
do i=1 to 3;
  if status01=. and status02=-4 and status03=-4 and
    oldsta01 gt 0 then do;
    status01=oldsta01;
  end;
  if status01=. and status02=-4 and status03=-4 and
    oldsta01 gt -4 and oldsta01 lt 0 then do;
    status01=-3;
  end;
  if status01=. and status02=-4 and status03=-4 and
    newsta01 gt 0 then do;
    status01=newsta01;
  end;
  if status01=. and status02=-4 and status03=-4 and
    newsta01 gt -4 and newsta01 lt 0 then do;
    status01=-3;
  end;
  if status02=. and status03=-4 and oldsta02 gt 0 then do;
    status02=oldsta02;
  end;
  if status02=. and status03=-4 and oldsta02 gt -4 and
    oldsta02 lt 0 then do;
    status02=-3;
  end;
end;

```

Appendix 3: Family Background and Formation Variable Creation

```

if status02=. and status03=-4 and newsta02 gt 0 then
  do;
  status02=newsta02;
end;
if status02=. and status03=-4 and newsta02 gt -4 and
  newsta02 lt 0 then do;
  status02=-3;
end;
end;

if id=1421191 then do;
  status01=1; status02=1;
end;

/* recoding status as we did in the first 3 rounds: 1 to 5 as
  stated above */
do i=1 to 3;
  if status(i)=1 then status(i)=5;
  else if status(i)=2 then status(i)=4;
  else if status(i)=3 then status(i)=1;
  else if status(i)=4 then status(i)=3;
  else if status(i)=5 then status(i)=4;
  else if status(i)=6 then status(i)=5;
  else if status(i)=7 then status(i)=5;
  else if status(i)=8 then status(i)=4;
  else if status(i)=9 then status(i)=2;
  else if status(i)=10 then status(i)=4; /*always double
    check these cases*/
  else if status(i)=11 then status(i)=3;
  else if status(i)=12 then status(i)=4;
  else if status(i)=999 then status(i)=-3;

/* picking up two missing cases from this questionnaire */
  else if dmonth(i) gt 0 then status(i)=2;
  else if status(i)=-4 then status(i)=-4;
  else if status(i)=-5 then status(i)=-5;
end;

if id=1462711 then status01=4;
if id=1278121 then status01=5;

/* sixth, the number of children ever born and residing in
  the household (tbiore) */
array biores(3) biores1 biores2 biores3;

do i=1 to 3;
  biores(i)=0;
  if status(i) eq 5 then biores(i)=1;
end;

tbiore=biores1+biores2+biores3;
if (mob01=-4 and mob02=-4 and mob03=-4) then
  tbiore=-4;
if (status01=-3 or status02=-3 or status03=-3) then
  tbiore=-3;
if mob01=-5 then tbiore=-5;

/* seventh, the number of children ever born and not
  residing in the household (tbionres) */
array bionres(3) bionres1 bionres2 bionres3;

do i=1 to 3;
  bionres(i)=0;
  if (status(i) eq 1 or status(i) eq 3 or status(i) eq 4) and
    status(i) ne 2 then bionres(i)=1;
end;

tbionres=bionres1+bionres2+bionres3;
if (mob01=-4 and mob02=-4 and mob03=-4) then
  tbionres=-4;
if (status01=-3 or status02=-3 or status03=-3) then
  tbionres=-3;
if mob01=-5 then tbionres=-5;

/* Last, sort created variables by birthdays, so that the first
  child listed is the oldest child. */
array smob(3) smob1 smob2 smob3;

do i=1 to 3; smob(i)=mob(i); end;
do i=1 to 3; if mob(i)=-3 then smob(i)=1000; end;

m1=-4; m2=-4; m3=-4;

/* consider the families with one child */
if (smob1>-4 and smob2=-4 and smob3=-4) or (smob1=-
  4 and smob2>-4 and smob3=-4) or (smob1=-4 and
  smob2=-4 and smob3>-4) then do;
  if smob1>-4 then m1=smob1;
  if smob2>-4 then m1=smob2;
  if smob3>-4 then m1=smob3;
end;

/* consider families with 2 children */
if (smob1>-4 and smob2>-4 and smob3=-4) then do;
  if smob1 ge smob2 then do;
    m1=smob2; m2=smob1;
  end;
  if smob1<smob2 then do;
    m1=smob1; m2=smob2;
  end;
end;

if (smob1=-4 and smob2>-4 and smob3>-4) then do;
  if smob2 ge smob3 then do; m1=smob3; m2=smob2;
  end;
  if smob2<smob3 then do; m1=smob2; m2=smob3; end;
end;

if (smob1>-4 and smob2=-4 and smob3>-4) then do;
  if smob1 ge smob3 then do; m1=smob3; m2=smob1;
  end;
  if smob1<smob3 then do; m1=smob1; m2=smob3; end;
end;

```

Appendix 3: Family Background and Formation Variable Creation

```
/* consider families with three children */
if (smob1>-4 and smob2>-4 and smob3>-4) then do;
  m1=min(smob1, smob2, smob3);
  m3=max(smob1, smob2, smob3);
end;

do i=1 to 3;
  if smob(i) ne m1 and smob(i) ne m3 then m2=smob(i);
end;

if smob1=-5 then m1=-5; if smob3=-5 then m3=-5;

array cv00bm(3) cv00bm1 cv00bm2 cv00bm3;
array cv00by(3) cv00by1 cv00by2 cv00by3;
array cv00mob(3) cv00mob1 cv00mob2 cv00mob3;
array cv00dm(3) cv00dm1 cv00dm2 cv00dm3;
array cv00dy(3) cv00dy1 cv00dy2 cv00dy3;
array cv00mod(3) cv00mod1 cv00mod2 cv00mod3;
array cv00stat(3) cv00sta1 cv00sta2 cv00sta3;

do i=1 to 3;
  if m1=smob(i) then do;
    cv00bm1 =bmonth(i); cv00by1 =byear(i);
    cv00mob1=mob(i); cv00dm1 =dmonth(i);

    cv00dy1 =dyear(i); cv00mod1=mod(i);
    cv00sta1=status(i);

  end;

  if m2=smob(i) then do;
    cv00bm2 =bmonth(i); cv00by2 =byear(i);
    cv00mob2=mob(i); cv00dm2 =dmonth(i);
    cv00dy2 =dyear(i); cv00mod2=mod(i);
    cv00sta2=status(i);
  end;

  if m3=smob(i) then do;
    cv00bm3 =bmonth(i); cv00by3 =byear(i);
    cv00mob3=mob(i); cv00dm3 =dmonth(i);
    cv00dy3 =dyear(i); cv00mod3=mod(i);
    cv00sta3=status(i);
  end;
end;

endsas;
```

NUMBER OF RESIDENCES SINCE AGE 12

Variables Created: CV_TTL_RESIDENCES

Variables Used

Name in Program	Question Name on CD
cvr1	CV_TTL_RESIDENCES
cvr2	CV_TTL_RESIDENCES
cvr3	CV_TTL_RESIDENCES
y35004	YHHI-3500
y36004	YHHI-3600
pubid	PUBID

This program calculates the number of residences in which youth has lived since age 12. In round 1, the variable was created with information from the parent interview. In round 2, information collected from the respondent was combined with the round 1 variable to update the previous information. Subsequent rounds continue to update the variable with respondent-provided data.

```

resid=-4;

/*R interviewed in R3 and R4*/
if cvr3=-1 or y36004=-1 then resid=-1; /*Refusal in R3 or R4*/
if cvr3=-2 or y36004=-2 then resid=-2; /*DK in R3 or R4*/
if cvr3=-3 then resid=-3; /*Parents not interviewed*/
if cvr3 ge 0 and y36004 ge 0 then resid=CVR3+y36004;
if cvr3 ge 0 and y35004=0 then resid=cvr3; /*Instead (if cvr3 ge 0 and y36004=-4 then resid=cvr3;) would also have
yielded same results.*/

/*R interviewed in R4 but not in R3 and also taking care of the cases when not interviewed in R2 as well*/

if cvr3=-5 then do;
  if cvr2=-1 then resid=-1;
  if cvr2=-2 then resid=-2;
  if cvr2=-3 then resid=-3;
  if cvr2 ge 0 and y36004 ge 0 then resid=cvr2+y36004;
  if cvr2 ge 0 and Y35004=0 then resid=cvr2;
  if cvr2=-5 then do;
    if cvr1=-1 then resid=-1;
    if cvr1=-2 then resid=-2;
    if cvr1=-4 then resid=-3;
    if cvr1 ge 0 and y36004 ge 0 then resid=cvr1+y36004;
    if cvr1 ge 0 and Y35004=0 then resid=cvr1;
  end;
end;

/*R not interviewed in R4*/
if y35004=-5 then resid=-5;

endsas;

```

